American Journal of Information Science and Computer Engineering
Vol. 1, No. 2, 2015, pp. 76-83
http://www.aiscience.org/journal/ajisce

Using Fuzzy ARTMAP for Symmetric Key Generation

John Mulopa, Siqabukile Ndlovu*, Kernan Mzelikahle, Thambo Nyathi

National University of Science and Technology, Computer Science Department, Bulawayo, Zimbabwe

Abstract

Neural cryptography deals with the problem of key exchange between two communicating neural networks using the mutual learning concept. It is the first algorithm for key generation over public channels which are not based on the number theory. The two networks exchange their outputs and the key between the two communicating parties is eventually presented in the final learned weights, when the two networks are synchronised. The security of neural synchronisation is put at risk if an attacker is capable of synchronising with any of the two parties during the training processes. However, the security of a cryptosystem is robust if the algorithm is strong and the keys are long, unpredictable, and random This research proposes use of two distant remote Adaptive Resonance Theory MAP (ARTMAP) architectures that are trained to learn from a unique data set and finally synchronise to same weights.

Keywords

Neural Networks, Cryptography, Cryptosystem, Synchronisation, Key Generation, Artmap

Received: April 20, 2015 / Accepted: May 17, 2015 / Published online: June 18, 2015

1. Introduction

The proliferation of portable devices; increased digital connectivity of networking devices; e-commerce applications; and the use of public infrastructure has increased the threat to information security. However, as technology advances and information management systems become more and more powerful, the problem of enforcing information security also becomes more critical [1]. More organisations are relying on the extensive use of the networked environment hence they are becoming vulnerable to security breaches [2]. Because of this vulnerability, there is need to secure information from unauthorised access by creating secure communication channels. One essential aspect for secure communications is cryptography [3].

Cryptography is the practice of hiding information. It involves the preparation of messages intended to be incomprehensible (encryption) to all except those who legitimately possess the means to recover the original information (decryption) [4]. Encryption transforms data into an unrecognisable format using a secret key, so that it is safe from sniffing. This data can be recovered only by decryption using the secret key. A common secret key could be created over a public channel accessible to any attacker. Very good cryptography gets its security by using incredibly long keys coupled with complex algorithms that are difficult to attack.

In data and telecommunications, cryptography is necessary when communicating over any untrusted medium, which includes just about any network, including the Internet. Cryptography not only protects data from theft or alteration, but can also be used for user authentication [4]. There are, in general, three types of cryptographic schemes typically used to accomplish these goals: secret key or symmetric cryptography, public-key or asymmetric cryptography, and hash functions [5]. In all cases, the initial unencrypted data is

* Corresponding author
E-mail address: jmulopa@gmail.com (J. Mulopa), siqabukile.sihwa@nust.ac.zw (S. Ndlovu), kernan.mzelikahle@nust.ac.zw (K. Mzelikahle), thambo.nyathi@nust.ac.zw (T. Nyathi)

referred to as plaintext. This plaintext is encrypted into ciphertext, which is the secret text transmitted to the recipient. With the correct key, the recipient is able to decrypt the ciphertext back to the initial plaintext [4] [6].

1.1. Artmap

Adaptive Resonance Theory (ART) was introduced by Grossberg as a means of describing how recognition categories are self-organised in neural networks. ART is a type of competitive learning network suitable for both pattern formation and recognition [7]. When an input pattern is adequately similar to one stored in the ART's long term memory, ART recognises the patterns as belonging to the same category, and modifies the stored category to accommodate new features of the current input pattern. An input pattern that is not closely similar to the stored category is stored in an uncommitted new category [7][8]. ART provides a mechanism by which the network can learn new patterns without forgetting or degrading old knowledge [8]. Fuzzy ART is a variation of ART system which uses fuzzy logic [9]. Fuzzy ART allows both binary and continuous input patterns. Fuzzy ART architectures perform unsupervised learning. In unsupervised learning also called self-organization training patterns of unknown classification are used, and there is no external teaching procedure [10]. An internal teaching function determines how network parameters are adapted based upon the nature of the input patterns [11].

1.2. Key Generation

The essential capability of cryptographic functions is random or pseudorandom number generation. The principal requirement for this capability is that the generated number stream be random and unpredictable. An important cryptographic function is strong pseudorandom number generation [4]. Keys are generated using algorithmic techniques for random number generation: pseudorandom number generator (PRNG), true random number generator (TRNG), and pseudorandom function (PRF). When a PRNG or PRF is used for a cryptographic application, the basic requirement is that if the seed value is unknown, an adversary is unable to determine the pseudorandom string. The general requirement of secrecy of PRNG or PRF is randomness and unpredictability of the seed. To achieve randomness, PRNG should be tested for uniformity, scalability, consistency, frequency tests, runs test, and Maurer's universal statistical test [4].

1.3. Artificial Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm inspired by the structure and functional aspects of biological neural networks to process information. It is configured for a specific application, such as pattern recognition or data classification, through a learning process [12]. An AAN applies computations that make it possible for computers to perceive, reason and act. It is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. For organisation purposes, processing elements known as neurons are grouped into layers. A normal neural network is composed of two layers with connections to the outside world: an input buffer where data is entered and an output buffer where the response of the network to the given input is stored. Layers between the input and output layers are named hidden layers. Neural networks have the ability to learn from examples [13]. It is sufficient to give some examples of the desired classification and the network takes care of the generalisation.

2. Problem Statement

Transfer of symmetric keys requires complex cryptography and the accurate independent generation of symmetric keys is an extremely difficult task to achieve. The primary security issue involves the generation of secret keys for cryptography which exhibit the characteristics of randomness and unpredictability. The current cryptosystems are being attacked by use of cryptanalysis, due to known public algorithms, key length, limited randomness and predictability in keys being used. What is required is to generate transitory keys that are random and unpredictable. The strength of the current cryptography algorithms can be enhanced by the use of neural networks. Neural networks learn by example and adapt their weights to different values in response to different inputs. This adaptive learning causes the weights to be extremely random and unpredictable. This novel approach can be used to generate keys that have the highest security requirements.

2.1. Proposed Solution

This paper seeks to adapt Fuzzy ARTMAP neural network architecture in the generation of symmetric keys for cryptography. The paper explores in detail the possibility of implementing neural networks in a novel symmetric or public-key cryptosystem. Novelty of this proposed solution lies in the use of autonomous learning behaviors of ART-based interacting neural networks. Once developed, the system is able to learn incrementally in real-time, non-stationary environments with minimum intervention from the system designer, computing or artificial intelligence experts.

The Fuzzy ARTMAP architecture has two fuzzy ART modules designated as ART_a and ART_b, as well as an inter-

ART module as shown in Figure 1. Inputs (a or I) are presented at the ART$_a$ module, while their corresponding outputs are presented at the ART$_b$ module. The inter-ART module includes a MAP field whose purpose is to determine whether the correct mapping has been established from inputs to outputs.

Fuzzy ARTMAP (FAM) operates in two phases, that is, the training phase and the performance or testing phase. In the training phase Fuzzy ARTMAP is presented with a set of input/output pairs and the network is trained. This is a learning phase in which FAM learns the output. In the testing phase the Fuzzy ARTMAP is presented with the input and the output obtained is compared with the actual output and the performance of Fuzzy ARTMAP is calculated.

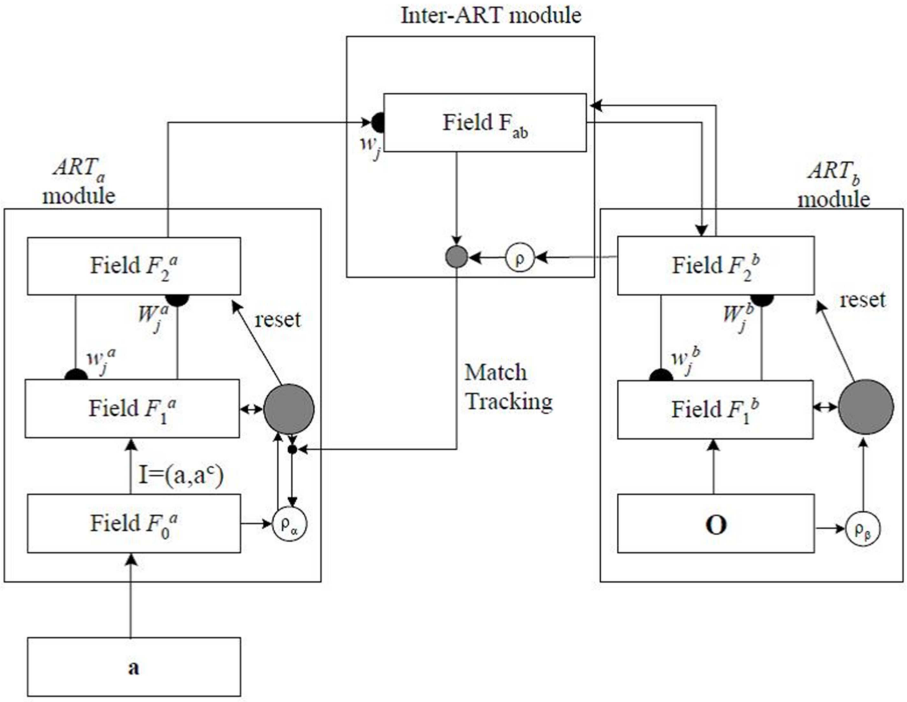


Figure 1. Fuzzy ARTMAP Architecture (Carpenter et al., 1987a).

2.2. Justification

In data and telecommunications, cryptography is necessary when communicating over any untrusted medium, which includes just about any network, particularly the Internet. Cryptography, then, not only protects data from theft or alteration, but can also be used for user authentication [4]. There are, in general, three types of cryptographic schemes typically used to accomplish these goals: secret key (or symmetric) cryptography, public-key (or asymmetric) cryptography, and hash functions. In all cases, the initial unencrypted data is referred to as plaintext. It is encrypted into ciphertext, which will in turn (usually) be decrypted into usable plaintext.

2.3. Related Work

The concrete proposal in [14] utilises two neural networks where each tries to learn from the other output, from common input vectors. The evidence for mutual synchronisation was found that when the network (teacher and student) has N weights including hidden neurons, then the training process during synchronisation requires an order of N examples to obtain generalisation abilities [14]. The average classification error decreases with an increase of training examples. Synchronisation of neural networks can be considered as the key generation in cryptography. The claimed basis for the security of the scheme in [14] is the proven fact that given fewer than some number of outputs of a parity machine with fixed weights for a random input, it is theoretically impossible to calculate these weights, and the case of changing weights seems to be even more complicated. However, the problem of computing the initial weights and that of finding the final weights are completely different and the attacker is only interested in the latter problem.

In 2010 the use of private inputs (queries) to the Tree Parity Machine (TPM) was proposed with empirical evidence for neural network synchronisation and cryptography [15]. The previous key-exchange protocols were extended by replacing random inputs with queries depending on the current state of the neural networks [15]. The queries are chosen alternatively according to their weight vectors. The query replaced the public input generated using pseudo-random number generator, as used in the basic neural cryptography. The synchronisation process now depended not only on the synaptic depth of TPM but also on queries. The only change in the basic neural key exchange algorithm is the public random input replaced with a query which is based on weight vector [14]. The algorithm works in two rounds. The first neural key exchange algorithm with query is applied till two networks are fully synchronised. The inputs (query) are visible to the attacker, but he cannot predict the TPM query (input) generated by either party as it is based on weights which are never exposed. After synchronisation, identical weight vectors are used as seed for pseudo

[16] proposed a new combined model for cryptography and steganography. The model uses the Hopfield Chaotic Neural Network (HCNN) for the cryptography which uses the chaotic trajectories of two neurons to produce main binary sequences for encrypting the plain-text. The model used also the Double Density Discrete Wavelet Transform (DD DWT) to embed the cipher-text into the audio cover. Experimental results show that the model is efficiency and secure against the most knows attacks.

In summary, neural networks (NN) are good for building very complicated cryptosystem which are difficult to break since the cracker has to know not only the topology of the NN and the key, but also the number of adaptive iterations and the final weights to be able to launch an attack. As a result of the analysis of the reviewed literature, it is clear that the current trend on the use of ANNs for cryptography is mainly focused on TPM, Chaotic Neural Networks (CNNs), and Layer Recurrent Neural Network (LRNN). This paper will focus on the use of fuzzy ART/ARTMAP. Evidence was found on the dissertation by [17] on “neural synchronisation and cryptography” about possibility to synchronise two neural networks that are TPM-based. Since sets of initial conditions exist on both TPMs, this will lead to identical weights and results [17]. This is the reason why it is possible to synchronise two neural networks without any interaction. Synchronisation of networks which do not interact with each other is much more difficult and takes longer time than performing the normal key-exchange [17].The purpose of this paper is to demonstrate the feasibility of symmetric key generation using extensive and exhaustive experimentation, and that the supervised learning fuzzy ARTMAP, is indeed a meritorious method of handling the generation of keys for symmetric cryptography with fast-learning to improve the synchronisation time.

2.4. Methodology

The proposed design is based on two independent Fuzzy ARTMAP neural network architectures. The proposed solution deals with the stated problem by synchronising the weights of the two independent fuzzy ARTMAP neural networks. This leads to identical weights and results [17]. The two Fuzzy ARTMAP networks are trained using the same training set. Once trained, the weights of the two ARTMAPs converge to identical weights that can be used by a cryptographic algorithm as a symmetric key for encryption and decryption, that is, the identical ARTMAP weight keys that are derived from a unique training set act as the secret key. The keys are generated by adapting the weights of the two neural networks. The keys are then deployed on a cryptography system using Data Encryption Standard encryption decryption encryption (DESede).

The research methodology used for this research is design science. Design science research is solution-oriented, which means it focuses on practical and theoretical knowledge by generating empirically tested design principles and methods [18,19]. For this paper the main objective is to develop new knowledge and design science research involves the creation of new knowledge through design of novel or innovative artifacts.

The methodology used for the development of the artefact is the Nested Loop Model of Neural Network Development Process [20, 21]. This model follows five stages and was used as follows:

Step 1: Network Requirements, Goals, and Constraints

The firm network requirements were done, this is the specification task. The goals were defined: To generate keys using two ARTMAP neural network architectures, synchronise the weights of two neural networks to be identical, and finally adapt the generated weights as keys. The output of the neural network was determined in terms of size of the weight vectors compared to the required key size. The constraints were on implementing the ARTMAP toolbox which is built on MATLAB environment. So we were forced to work under the MATLAB environment and using interfaces from MATLAB R2013a. However, to build good interfaces we had to use Java which we had to interface with MATLAB, and determination of the main language (which calls which), had to be done.

MATLAB R2013a and Java were loaded on Linux platform (Centos 6.5). MATLAB R2013a includes an extensive toolbox of numerical analysis algorithms, so the programming effort often involves implementing the mathematical models, characterizing the input data and applying the available numerical algorithms. MATLAB R2013a can handle native Java objects in its workspace and can directly call Java. Java is well suited to graphical user interface development. Java is powerful with cryptography classes and those were used to develop a block cipher algorithm. The Fuzzy ARTMAP toolbox designed by Aaron Garrett was loaded into MATLAB R2013a for testing the algorithm.

Step 2: Data Gathering and Preprocessing

Data gathering involved assembling all the data that would be used in the training of the ARTMAP neural network. Training sets were obtained from MATLAB datasets. This helped with the determination of the variable weight vector size as we increased the training sample into the algorithm, to assist in design of the block cipher. The preprocessing was done using MATLAB nntool to convert the training data set into an electronic format that is consistent with MATLAB R2013a.

Step 3: Training and Testing Loops

The architecture only was trained for deep understanding of the algorithm. The intra-paradigm architectural parameters were varied to investigate the changes in the layer nodes and weights. The neural network was tested by varying the initial parameters. The weights were adapted and converted into keys. Once trained the keys were tested on a cipher and two remote systems tested using two sessions of MATLAB to test the encryption process.

Step 4: Network Deployment

The final neural net was deployed as a module within the MATLAB program. MATLAB R2013a was linked to the networking module, neural network module and the cipher module to create automatic code generation. After the network was trained, source code to compile within the host application was created using MATLAB and Java including the block cipher program. ANN data was loaded into MATLAB to exercise the network and perform encryption and decryption. All deployment effort was documented and recorded using manually or automatically generated source code. Finally, the required data pre- and post-processing requirements for input and output parameters was explicitly defined, along with analysis of possible error conditions the ANN and block cipher will cause.

Step 5: Independent Testing and Verification

When all the constraints were met and all activities documented, the system was tested for verification. We ascertained that the objectives have been met and evaluated any possible conclusions arrived at. A critical review of all the phases of development effort was reviewed for accepted procedures. If the phases were found deficient the program was stepped back to data gathering phase.

2.5. Implementation

The solution comprises a Fuzzy ARTMAP neural network adapted for training and classification using a data set. The ARTMAP is used to classify the data set and the final weights are adapted as keys for the cipher module. The solution also includes the cipher module that will perform encryption and decryption of text. Also included is the client-server module to handle the transmission control protocol for the creation of the communication channel. The ARTMAP neural network is set up with a number of initial parameters including the cancer data set: vigilance parameter, learning rate, number of epochs, and maximum number of categories, input neurons, and classes. The solution is setup on the same machine using two MATLAB sessions connected via a TCP/IP communication channel.

The solution is made up of three subsystems levels: user level, neural network level, and cryptosystem level as indicated in Figure 2.

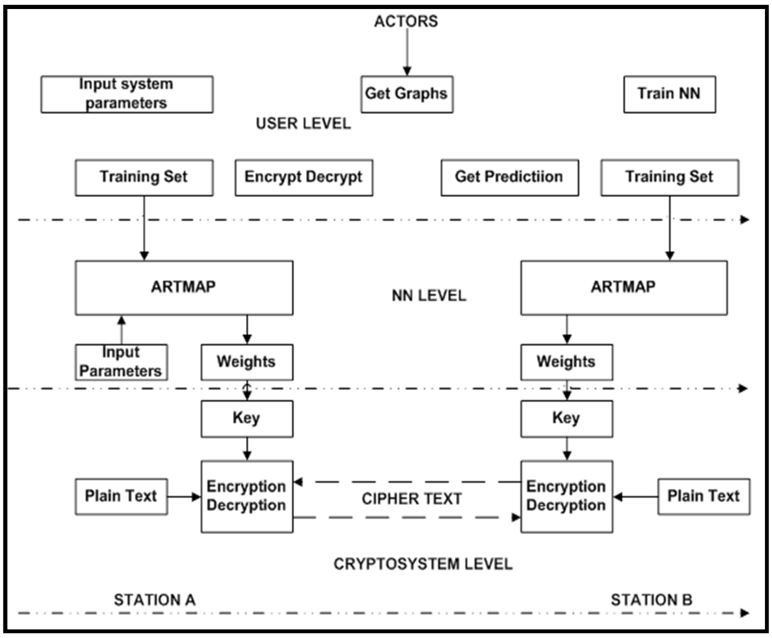


Figure 2. Block diagram of solution.

Input is fed through the interface and is pre-processed using complement coding. All input values of input vector must be within the range [0.1]. If not the inputs are analogue and the input vector should be normalised. Component coding is required to preserve the amplitude information. The neural network level (NN LEVEL) performs various actions on the input data and requests and sends the data to the cryptosystem or back to the user. This level consists of input data pre-processing, normalisation, Neural Network training, and weights vector generation. The preprocessed input vectors are classified by the ARTMAP architecture into categories. This classification increases the weight matrix. As the training sample size increases new categories are classified hence the variables change in the size of the weight matrix (categories). If a match criterion in ARTMAP determined by a vigilance parameter is met by comparing with the activation function caused by input vectors, then a category is created. The cryptosystem level contains the cryptography algorithm that will encrypt and decrypt data input by the user. The user inputs system parameters, initialises the neural network and selects the training data set. The input training set is used for training, testing, and validating the neural network. The training and reconfiguration of the neural network is iterative as long as the vigilance test fails i.e. the degree of match between input to the prototype (already memorised categories) is very large. A failed test means the input is not close to the already memorised categories, so a new category will be created. a passed vigilance test will cause the network to be saved along with the corresponding weight matrix. The weight matrix is converted into a key which is used for encryption and decryption using the DES algorithm. The plain text is encrypted using the generated key. The user can request for statistical information, graphs, key, weights, etc via the user interface. The user trains the neural network using the training set and chosen Fuzzy ARTMAP system parameters. The trained ARTMAP neural network presents a weight vector matrix that is adapted into a key. This key is used for encryption of the plaintext message and decryption of the ciphertext. The architecture aims at coordinating parameters that advantageously facilitates balancing the keys for both fuzzy ARTMAP networks to be similar through training of the two networks using a unique training set. After synchronising the two non-interacting neural networks, a pair of secret-keys having the same values can be obtained individually at both sides i.e. sender and receiver independently generate an identical key. This removes the need for the two networks to exchange the key.

2.6. Testing and Results

Two MATLAB sessions were started on one local machine and a client server transaction control protocol/internet protocol (TCP/IP) communication channel setup between the two MATLAB sessions to simulate two remote host systems. A communication channel was setup between the client and the server MATLAB sessions. When the communication channel was established, training of the neural network ensued at both ends. Important parameters were input in the ARTMAP graphic user interface to test the changes in the weights as the input parameters were varied. The changes in the weights were done using the command line to enable easy analysis of the changes in the parameters. Two scripts were created to enable testing of the two remote sessions: one for Server and the other for Client. Another script was created to test vigilance parameter against the creation of new categories as shown in Table 1 and Table 2. The ARTMAP was setup as per parameters in Table 1 below.

Table 1. ARTMAP Test parameters.

Parameter	Value
Vigilance	0.1-1.0
numClasses	2
Learning rate	1.0
numEpochs	10
maxNumCategories	150

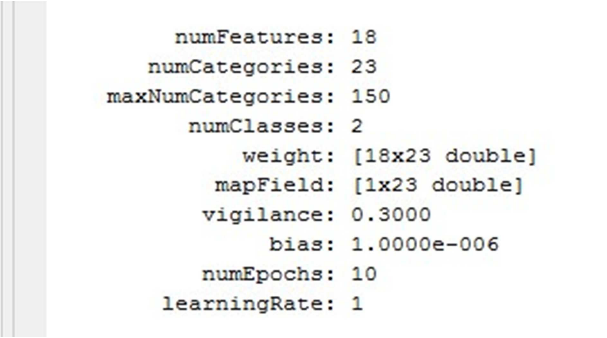
The results in Table 2 show that ARTMAP classifies a data set poorly when the vigilance parameter is very small by producing coarse categories. The number of epochs required to classify the data set is increased. Training with high vigilance value causes fuzzy ARTMAP to generate small prototypes, thus making classification that are more precise, but requiring more neurons to perform classification hence changes in the weight matrix size. A low vigilance value commits fewer neurons, but creates a rougher classification with large prototypes.

Table 2. ARTMAP Test Results.

Vigilance =1.0	Classification Time	Runs Number	Epochs	Categories
	3.9712	1	2	150
	3.7837	2	2	150
	3.7697	3	2	150
Vigilance =0.9	2.0269	1	2	150

Vigilance =1.0	Classification Time	Runs Number	Epochs	Categories
	1.9829	2	2	150
	1.9155	3	2	150
Vigilance =0.8	0.7787	1	2	126
	0.7882	2	2	126
	0.7813	3	2	126
Vigilance =0.7	0.7959	1	3	85
	0.8121	2	3	85
	0.8034	3	3	57
Vigilance =0.6	0.7685	1	4	57
	0.7686	2	4	57
	0.7828	3	4	57
Vigilance =0.5	1.1168	1	7	45
	1.1337	2	7	45
	1.1060	3	7	45
Vigilance =0.4	0.5156	1	4	31
	0.5223	2	4	31
	0.5112	3	4	31
Vigilance =0.3	0.6577	1	6	23
	0.6239	2	6	23
Vigilance =0.2	0.6019	1	6	17
	0.5684	2	6	17
Vigilance =0.1	0.4305	1	5	14

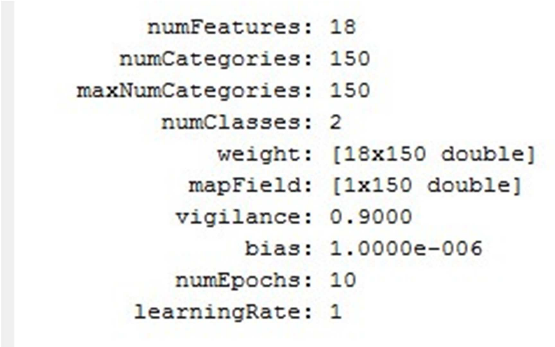
Though the time required to classify is small. As the vigilance parameter value increases so does the number of categories created to classify the output. The epochs required to completely classify a large data set decreases. The length of the weight vector is always proportional to input by categories classified. The randomness of the key is with the increase in the vigilance parameter and the categories classified as shown in the Fig 3 and Fig 4 below:



```
      numFeatures: 18
    numCategories: 23
 maxNumCategories: 150
       numClasses: 2
           weight: [18x23 double]
         mapField: [1x23 double]
        vigilance: 0.3000
             bias: 1.0000e-006
        numEpochs: 10
     learningRate: 1
```

Figure 3. Network Created at Vigilance=0.3.

Fig 3 above shows that the weight vector created at vigilance =0.3 is 18x23 double vector and the randomness of the weights is low. Low vigilance means the classification or degree of match is very loose. As we increase the vigilance close to one we realise that ARTMAP increases its level of classification as shown by Fig 4. The weight vector size increased tremendously 18x150. The randomness of the weight vector values increases as the vigilance increases.



```
      numFeatures: 18
    numCategories: 150
 maxNumCategories: 150
       numClasses: 2
           weight: [18x150 double]
         mapField: [1x150 double]
        vigilance: 0.9000
             bias: 1.0000e-006
        numEpochs: 10
     learningRate: 1
```

Figure 4. Network Created at Vigilance=0.9.

From the times shown on Table 2 it shows that ARTMAP system learns very fast using a few epochs (a few input patterns). It achieves these properties by using an internal controller (vigilance parameter) that conjointly maximises predictive generalization and minimizes predictive error.

From the results we can conclude that the ARTMAP system learns very fast by using an internal controller (vigilance parameter) that conjointly maximises predictive generalisation and minimises predictive error.

Connecting for first time using different input training parameters for neural network causes the neural network to fail to decrypt the text. But repeated trials cause the neural network to start encrypting and decrypting.

The developed system was evaluated against the following:

1) Accuracy of weight vectors

2) Functionality. Test if the system is functioning properly.

3) Performance. Parameters were varied and synchronization time determined.

3. Conclusion

From results shown in section 2.6 we can conclude that ARTMAP is a very efficient classification algorithm. With high vigilance ARTMAP can take as minimum as two epochs to reach equilibrium. The changes in the vigilance also changed the weight vector matrix size and made the values very random as categories increased. Due to use of one data set to train neural network classification was done efficiently resulting in the constant weights and the key memorised. Once the key has been generated one can encrypt and decrypt as they like without re-entering the key. This can improve the security risk that comes with the exchange of symmetric keys. Changes to the data set cause a change in the weights and hence the key. ARTMAP's fast learning typically leads to different adaptive weights and recognition categories for different orderings of a given data set used, even when the overall predictive accuracy of all simulations is similar.

References

[1] Modaressi, A. R., S. Mohan, 2000. Control and management in Next-Generation Networks: challenges and opportunities. IEEE Commun. Mag., 38:94-102.

[2] Oates J.B., (2011). Researching Information Systems and Computing.Sage Publications Inc, California, USA.

[3] Pratap S., Harvir S. Cryptography in structure adaptable digital neural networks. National monthly referred journal of research in science and technology. Vol 1, issue 12: ISSN 2277-1174.

[4] Stallings W., (2010) Cryptography and Network Security: Principles and Practice, (5th Edition), Prentice Hall, 2010.

[5] Kelsey J., B. Schneier, D. Wagner, C. Hall: Cryptanalytic Attacks on Pseudorandom Number Generators.

[6] Santhanalakashmi S, K. Sangeeta, G.K. Patra (2012). Design of stream Cipher for Text Encryption using Soft Computing based Techniques. International Journal of Computer Science and Network Security, Vol.12 No. 12.

[7] Carpenter, G.A, M.N Gjaja, S. Gopal, N Markuzon, C.E. Woodcock. (1996). ART and ARTMAP Neural Networks for Applications: Self-Organizing learning, Recognition, and Prediction. Boston University, Technical Report CAS/CNS TR-96-009. Massachusetts 02215 USA.

[8] Carpenter, G.A., Grossberg, S., and Reynolds, J.H. (1991a). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. Neural Networks, 4, 565-588.

[9] Carpenter, G.A. and Grossberg, S. (1992). Fuzzy ARTMAP: Supervised learning, recognition, and prediction by a self-organizing neural network. IEEE Communications Magazine, 30, 38-49.

[10] Carpenter, G.A., Grossberg, S., Markuzon, M., Reynolds, J.H., and Rosen, D.B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Transactions on Neural Network, 3,698-713.

[11] Carpenter, G.A., Grossberg, S., and Rosen, D.B. (1991b). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks, 4, pp. 759-771.

[12] Shukla N., A. Tiwari. An Empirical Investigation of Using ANN Based N-State Sequential Machine and Chaotic Neural Network in the Field of Cryptography‖, Global Journal of Computer Science and Technology Neural & Artificial Intelligence, Vol. 12, Issue.10,No. 1,17-26, 2012.

[13] Prabakaran N., P. Loganathan, P. Vivekanandan, 2008. Neural cryptography with multiple transfers functions and multiple learning rule. International Journal of soft computing 3 (3):177-181.

[14] Kanter I., Kinzel W, Kanter E. Secure Exchange of information by synchronisation of neural networks, Europhys, Lett. 57, 141, 2002.

[15] Revankar P, W.Z. Gandhare, D Rathod. Private Inputs to Tree Parity Machine. International Journal of Computer Theory and Engineering, Vol. 2, No. 4, August, 2010: 1793-8201.

[16] Geetha B, E. Vani, V. Prasad. A Hybrid Model for Secure Data Transfer in Audio Signals using HCNN and DD DWT‖, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No.7, 202-208, 2013.

[17] Ruttor A., G. Reents, W. Kinzel, 2004. Synchronisation of random walk with reflecting boundaries. J. Phys. A: Math.Gen, 37:8609 [condmat/0405369].

[18] Vaishnvai V., W. Kuechler (2004). Design Science Research in Information Systems, January 20,2004; last updated: October 2013. Available: http://www.desrist.org/design-research-in-information-systems/ (Accessed 20 February 2015)

[19] Hevner, A. (2007). A three-cycle view of design science research, Scandinavian Journal of Information Systems 19 (2), pp. 87–92

[20] Bedford D.F, G. Morgan, and J. Austin, "A Draft Standard for the Certification of Neural Networks in Safety Critical Systems," Artificial Neural Networks in Engineering, November 1996

[21] Taylor B.J.,(2006). Methods and Procedures for the Verification and Validation of Artificial Neural Networks. Springer, Fairmont, USA.